Technical Report JPL-MLTR-02-1

AN ARROW-BASED NOTATION FOR SIMULATING SIGNAL TRANSDUCTION NETWORKS WITH CELLERATOR

Applications Note

to be submitted to *Bioinformatics*

Bruce E. Shapiro* and Eric D. Mjolsness

Machine Learning Systems Group, Jet Propulsion Laboratory, California Institute of Technology, Mail Stop 126-347, 4800 Oak Grove Drive, Pasadena, CA 91109

Running Head: Automated Equation Generation with Cellerator

Key words: Cellerator, Automatic Equation Generation, Signal Transduction, Computer Algebra, SBML

*To whom correspondence should be addressed

ABSTRACT

Summary: Cellerator is designed to simulate single and multi-cellular signal transduction networks. Interactions, specified by a compact, optionally palette-driven, arrow based notation, are symbolically translated into differential equations using a computer algebra system (Mathematica) that can be subsequently solved numerically and/or output in a variety of formats including SBML, C, and FORTRAN. Cellerator simulations can be run automatically or with full user intervention and allow complete manual modification of chemical and mathematical equations at all levels of simulation.

Availability: http://www-aig.jpl.nasa.gov/public/mls/

Contact: bshapiro@jpl.nasa.gov

Cellerator is a Mathematica package designed to facilitate biological modeling via automated equation generation. Cellerator was designed with the intent of simulating at least the following essential biological processes: (a) signal transduction networks (STNs); (b) cells that are represented by interacting signal transduction networks; and (c) multi-cellular tissues that are represented by interacting networks of cells that may themselves contain internal STNs. These processes combine to form an obvious hierarchy that can be further subdivided for notational simplicity (e.g., STNs as elements of STNs, and so forth).

Signal transduction networks are specified using an arrow-based language (Shapiro, Levchenko and Mjolsness, 2001) to represent interactions between various chemical species, up to and including simplified representations of transcriptional regulation (Fig. 1). The general input canonical form is

$$\{rlist \ \begin{matrix} catalyst \\ arrow \\ catalyst \end{matrix} \ rlist \ , \ clist \ \}$$

where *rlist* is a list of reactants (e.g., $A+B+C$); *arrow* is one of the arrows in table 1 (e.g., $\rightarrow, \rightleftharpoons, \mapsto$), *catalyst* is an optional species that catalyzes the reaction (such as an enzyme; the upper and lower catalysts affect forward and reverse reactions, respectively); and *clist* is an optional list of comma-delimited rate constants (either symbolic names or values). Reactions can also be specified by clicking on the appropriate buttons on a palette (Figure 1). After being collected the reactions are then symbolically translated into differential equations. The Cellerator implementation also allows explicit output description at each level so that "power-users" can modify the equations at any stage desired. Output is produced in a variety of formats: as Mathematica ODEs, in C, FORTRAN, SBML (Hucka, 2001), MATHML, or HTML. If desired, the user can optionally solve the equations numerically. The present interface is a

palette-driven Mathematica notebook; future enhancements include GUI, Java and web-based interfaces.

Multicellular systems are represented by graphs containing a list of *nodes*, a list of *links*, and a *lineage tree*. Nodes represent cells; links represent intercellular interactions; and the lineage tree records the familial history of cell birth. Cell division occurs (optionally) whenever a specified variable) passes a threshold. New cells are added to the graph when cells divide.

In additional to the various forms of direct transcriptional regulation illustrated in table 1, large genetic regulatory networks can be represented by a generalization of the connectionist model proposed previously (Mjolsness et. al. 1991). Suppose that a cell contains n chemical species whose concentrations are denoted by $v_a,\ a = 1,2,...,n$. Then the basic equation is

$$\tau_a \dot{v}_a = g(u_a + h_a) - \lambda_a v_a + \tau_a \dot{v}_{a,Cellerator}$$

where $\tau_a,\ h_a$ and λ_a are constants, $\dot{v}_{a,Cellerator}$ is the sum of the terms generated from table 1, and

$$u_a = \sum_b T_{ab} v_b + \sum_{i \in Nbrs} \Lambda^i \sum_b \hat{T}_{ab} \hat{v}_b^i + \sum_{i \in Nbrs} \Lambda^i \sum_b \sum_c \tilde{T}_{ac}^{(1)} \tilde{T}_{cb}^{(2)} v_c \hat{v}_b^i$$

where T_{ab} is a connection strength matrix giving the effect of concentration v_b on concentration v_a; i is an index that runs over all neighboring cells; Λ^i gives the geometric connectivity between the current cell and neighboring cell i; $\hat{v}_b^i$ is the concentration of species v_b in neighboring cell i; $\hat{T}_{ab}$ is a connection strength matrix that gives the effect of $\hat{v}_b^i$ on v_a; v_c is the receptor concentration; v_b^i is concentration of ligands excreted by neighbor i; $\tilde{T}_{cb}^{(2)}$ is the connection strength for excitation of receptor *c* due to ligand *b;* $\tilde{T}_{ac}^{(1)}$ is the connection strength

for production of protein *a* via receptor *c* activation; and . The $g(u)$ is a monotonic saturating function such as $g(x) = 0.5(1 + x / (x^2 + 1)^{1/2})$ (Reinitz et. al. 1995).

Cell growth is represented by associating a "spring potential" with each link. A gradient descent towards local minimum is incorporated by adding equations of the form $\dot{\mathbf{x}}_i = -\nabla V_{ij}$ where the potential function for the ith node is

$$V_{ij} = \frac{1}{2}\sum_j k_{ij} c_{ij} [(|\mathbf{x}_i - \mathbf{x}_j| - d_{ij})^2 - \mu]$$

where the sum is taken over all nodes *j* that are linked to node *i*, $\mathbf{x}_i$ are (vector) node locations, k_{ij} are interaction strengths, and d_{ij} give the desired separation between the nodes. The potential gets "turned off" (set to zero) when the interaction distance becomes too large ($d_{ij} > d$ for some constant *d*).

In the past it has been necessary to manually translate chemical networks from cartoon-diagrams to chemical equations and thence to ordinary differential equations. This process is tedious and highly error prone, and impractical for all but the simplest of systems because of the combinatoric increase in the number of equations with the number of chemical species. Cellerator provides a framework for generating, translating, and numerically solving a potentially unlimited number of biochemical interactions.

ACKNOWLEDGEMENT

Portions of this research were carried out by the Jet Propulsion Laboratory, California Institute of Technology, under contract with NASA; by the Whittier Foundation; the ERATO Kitano Symbiotic Systems Project; the NASA Biomolecular Systems Research Program; and the Caltech President's Fund. E. Meyerowitz, B. Wold, A. Levchenko, and H. Bolo Uri provided valuable suggestions during Cellerator development.

REFERENCES

Elowitz, M. B. and Leibler, S. (2000) A synthetic oscillatory network of transcriptional regulators, *Nature*, 403:335-338.

Hucka M., Finney, A., Sauro, H., and Bolouri, H. (2001) Systems biology markup language (SMBL) level 1: Structures and facilities for basic model definitions. 2 March 2001, on the World Wide Web at http://www.cds.Caltech.edu/erato.

Mjolsness, E., Sharp, D.H., and Reinitz, J. (1991) A connectionist model of development. *J. Theor. Biol.,* **152**: 429-453.

Reinitz, J, Mjolsness, E, and Sharp, D. (1995) Model for cooperative control of positional information in *Drosophila* by bicoid and maternal hunchback, *J. Exp. Zool.,* 271:47-56.

Shapiro, B.E., Levchenko, A., and Mjolsness, E. (2001) Automatic model generation for signal transduction a to MAP-Kinase pathways, in *Foundations of Systems Biology*, ed. H. Kitano, MIT Press.

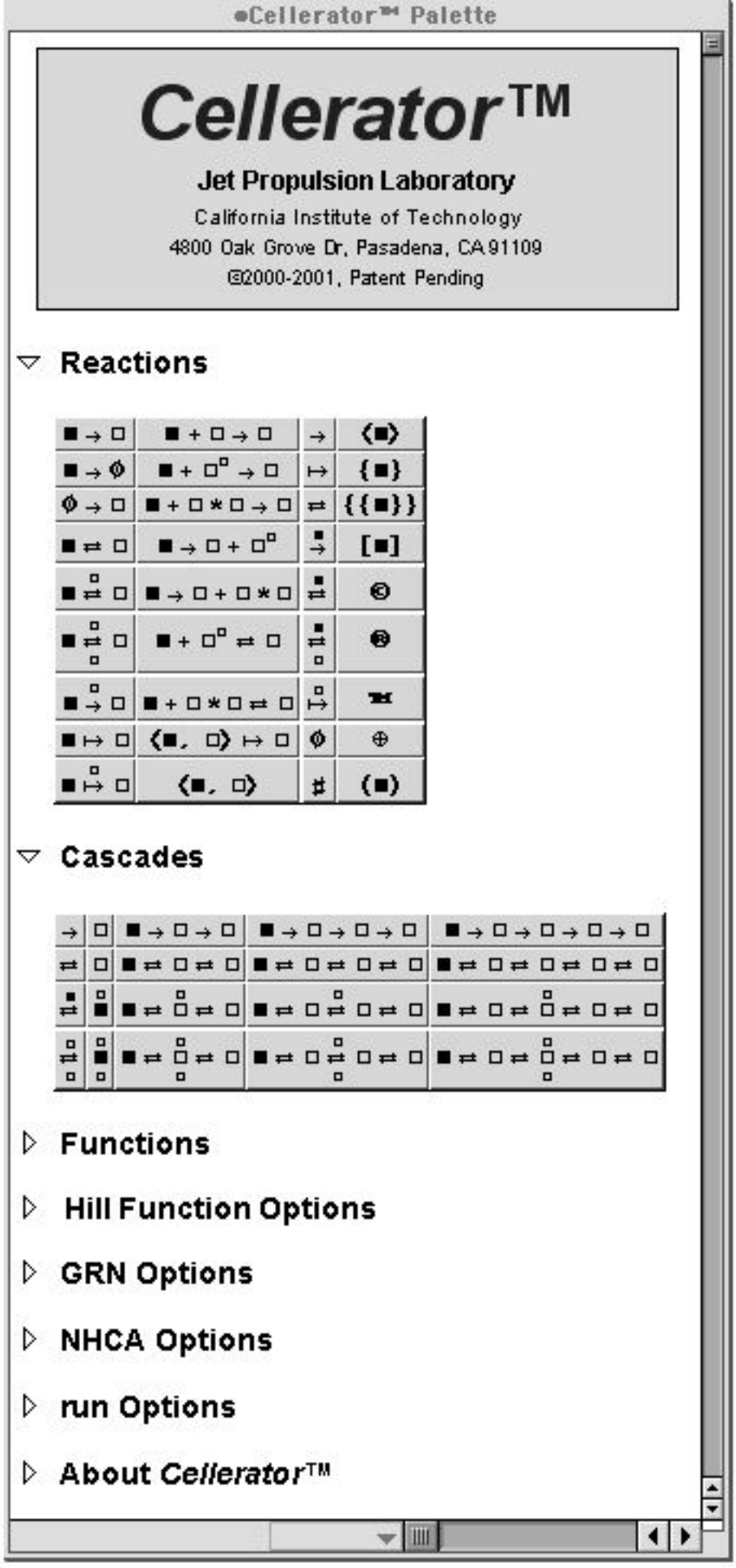
Shapiro, B.E., and Mjolsness, E. D. (2001) Developmental simulations with Cellerator, *Proceedings of the Second International Conference on Systems Biology*, pp. 342-351, also available on the World Wide Web at http://www.icsb2001.org/toc.html.

FIGURE CAPTION

Figure 1. Top: The Cellerator palette. Bottom: Typical notebook interface, using Cellerator to simulate the repressilator (Elowitz and Leibler, 2000).

Table 1. Cellerator arrow notation.

Cellerator Arrow	ODE Interpretation
$\mathtt{S} \to \mathtt{P}$	$\dot{S} = -\dot{P} = -k$
$\mathtt{A} + \mathtt{B} \to \mathtt{C}$	$\dot{A} = \dot{B} = -\dot{C} = -kAB$
$\mathtt{A} + \mathtt{B}^{\mathtt{n}} \to \mathtt{C}$	$\dot{A} = \dot{B} = -\dot{C} = -kAB^n$
$\mathtt{A} \rightleftharpoons \mathtt{B}$	$\dot{A} = -\dot{B} = -k_f A + k_r B$
$\mathtt{A} + \mathtt{B} \rightleftharpoons \mathtt{C}$	$\dot{A} = \dot{B} = -\dot{C} = -k_f AB + k_r C$
$\varnothing \to \mathtt{A}$	$\dot{A} = k$
$\mathtt{B} \to \varnothing$	$\dot{B} = -kB$
$\mathtt{S} \overset{\mathtt{E}}{\rightleftharpoons} \mathtt{P}$	$\dot{S} = -a \cdot E \cdot S + d \cdot S, \;\; P\cdot = k \cdot (SE)$ $\dot{E} = -a \cdot E \cdot S + (d + k) \cdot (SE) = -(\dot{SE})'$
$\mathtt{S} \underset{\mathtt{F}}{\overset{\mathtt{E}}{\rightleftharpoons}} \mathtt{P}*$	Equivalent to $\mathtt{S} \overset{\mathtt{E}}{\rightleftharpoons} \mathtt{P}$ and $\mathtt{P} \overset{\mathtt{F}}{\rightleftharpoons} \mathtt{S}$
$\mathtt{S} \overset{\mathtt{E}}{\to} \mathtt{P}*$	$\dot{S} = -k \cdot E \cdot S = -\dot{P}$
$\mathtt{S} \overset{\mathtt{E}}{\mapsto} \mathtt{P}*$	$\dot{P} = \dfrac{(k + vE)S^n}{K^n + S^n} = -\dot{S}$
$\mathtt{A} \mapsto \mathtt{B}$ (Hill)	$\dot{B} = r_0 + \dfrac{(r_1 + \sum_{i=1}^{p} v_i A_i)^n}{K^n + (r_1 + \sum_{i=1}^{p} v_i A_i)^n}$
$\mathtt{A} \mapsto \mathtt{B}$ (GRN)	$\dot{B} = R\left[1 + \exp\left(-\sum_{i=1}^{p} T_i A_i^{n_i} + h_i\right)\right]^{-1}$
$\mathtt{A} \mapsto \mathtt{B}$ (NHCA)	$\dot{B} = \dfrac{1 + (\sum_{i=1}^{p} T_i^{+} A_i^{n_i})^m}{k(\sum_{i=1}^{p} T_i^{-} A_i^{n_i})^m + (\sum_{i=1}^{p} T_i^{+} A_i^{n_i})^m}$




```
s = interpret[{
    {PZ ↦ X, hill[vmax → α1, nhill → n, basalRate → {α0, α}]},
    {PX ↦ Y, hill[vmax → α1, nhill → n, basalRate → {α0, α}]},
    {PY ↦ Z, hill[vmax → α1, nhill → n, basalRate → {α0, α}]},
    {X → ϕ, 1}, {Y → ϕ, 1}, {Z → ϕ, 1},
    {ϕ →(X) PX, β}, {ϕ →(Y) PY, β}, {ϕ →(Z) PZ, β},
    {PX → ϕ, β}, {PY → ϕ, β}, {PZ → ϕ, β}
   }];
Print[s];
r = run[s,
   rules → {α → 250, α0 → 0, α1 → 0, n → 2.1, β → 5},
   initialConditions → {PX[0] == 5, PY[0] == 10, PZ[0] == 15},
   plotVariables → {PX, PY}, plotColumns → 2, timeSpan → 30];
```

$$\left\{\left\{PX'[t] == -\beta\, PX[t] + \beta\, X[t],\right.\right.$$
$$PY'[t] == -\beta\, PY[t] + \beta\, Y[t],\ PZ'[t] == -\beta\, PZ[t] + \beta\, Z[t],$$
$$X'[t] == \alpha 0 + \frac{\alpha + \alpha 1\, PZ[t]^n}{1 + PZ[t]^n} - X[t],\ Y'[t] == \alpha 0 + \frac{\alpha + \alpha 1\, PX[t]^n}{1 + PX[t]^n} - Y[t],$$
$$\left.\left. Z'[t] == \alpha 0 + \frac{\alpha + \alpha 1\, PY[t]^n}{1 + PY[t]^n} - Z[t]\right\},\ \{PX, PY, PZ, X, Y, Z\}\right\}$$